Modeling Cross-Linguistic Phonological Patterns using Emergent Communication

Rujul Gandhi

24.918

May 15, 2022

1 Introduction

Remarkable cross-linguistic similarities in how natural language phonologies have evolved presents an interesting modeling problem. The convergence upon 'universals' in the segment inventory or in sequential constraints could be driven by a variety of factors - innate human tendency, perceptual clarity, articulatory ease, or imitation of others being some proposals. Past models and simulations of this evolution process have focused on agents imitating each other to develop a phonology, but the pressure of needing to communicate efficiently about the world does not factor into these models. This project presents a communication-based approach to the evolution of phonological patterns by proposing experiments in an emergent communication setting. This is a paradigm where multiple neural networks are trained to achieve related goals (which may be collaborative or competitive), and are allowed to send 'communication' tokens to each other. The content and nature of this communication can be studied. In this set of experiments, the networks are built to communicate through a sequence of phonemes from a pre-defined phone set.

It is intended to be a proposal for modeling the evolution of phonological universals, and studying how patterns formed by statistical models optimizing for efficient communication are similar to, or differ from, patterns attested in human languages today.

2 Theoretical Motivation

This study is motivated by the general effort of the field of linguistics to develop a theoretical model of human communication. There are two aspects to this - forming hypotheses and testing them out. Computational models and simulations are one way to test hypotheses, along with fieldwork or experiments with human speakers. A computational model or simulation tries to isolate aspects of human communication and reproduce them, away from the complexities of the human mind. The contribution of this modeling process is that the disparities and similarities between the model results and parallel studies in humans can lead to insights about what modeling assumptions were correct.

In some ways, phonology lends itself well to this sort of modeling. We are able to develop a clear modeling system where we can control the number of phones, noise in the environment, and how phones are defined. There are also widely attested patterns in human languages, available in surveys and databases containing hundreds of languages - such as the UCLA Phonological Segment Inventory Database (UPSID) (Maddieson, 1984) or the Stanford Phonology Archive (Crothers, Lorentz, Sherman, & Vihman, 1979). These databases makes it easier for us to isolate phenomena we care about and verify our findings.

On the other hand, in some ways modeling phonology is a cloudy task. Specifically, most theoretical models of phonology treat phonemes as collections of categorical features. However, sound in general is much more scalar or gradual, and modeling the acoustic properties of sounds in a way that also gives us information about traditionally defined ‘phonemes’ can be tricky. Some of the work in the next section addresses this through various ideas and generalizations.

3 Background

This section presents current concepts and frameworks that will be referenced throughout this paper. First, we look at typological studies of phonology and discourse about ‘universals.’ Then, some past approaches to modeling phonology are studied. Finally, this section provides an overview of emergent communication and its potential applications to linguistics.

Phonological Universals In an extensive survey and analysis of the concept of universals in phonology, (Hyman, 2008) suggests that two key things are absolutely necessary for the ‘phonology’ for a language. Specifically, Hyman claims that a language without a phonology of some sort would lack a ‘fixed inventory of distinctive segments’ or any ‘sequential constraint(s) on segments.’ Hyman also contrasts ‘descriptive’ universals with ‘analytic’ universals, in which the latter are more theory-dependent and the former can be stated broadly about languages with minimal dependence on a particular theoretical framework. This project specifically probes the ‘descriptive’ universals.

First, we consider the fixed inventory of contrastive segments in a language (henceforth, ‘segment inventory’). The UPSID survey compiles contrastive segment inventories for 451 natural languages. While the language with the largest segment inventory in this set has an inventory of 141 phonemes, the average number of sounds in a phoneme inventory is just between 20-37 (Maddieson, 1984). The selection of which sounds are contrastive is not at random, with Maddieson in 1984 doing an analysis of 317 languages and finding that just within consonants, over 90% of the surveyed languages included some nasal and stop sounds (/m/, /n/, /t/, /k/) with that percentage decaying down to just about 20% of languages including the $25th$ most common

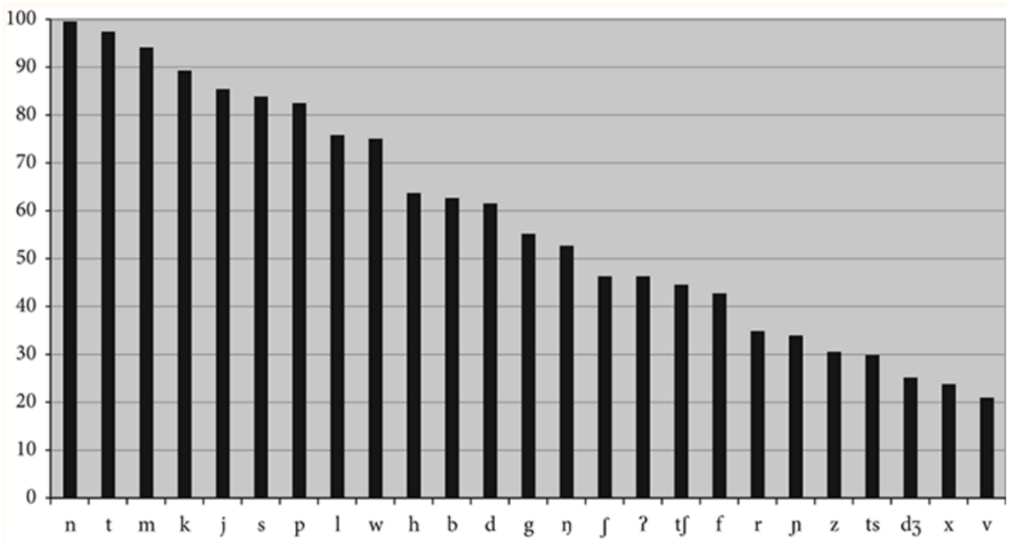


Figure 1: From Maddieson (1984): Percentage of languages in the surveyed set that contained each of the 25 most common consonants.

consonant sound /v/. These results from Maddieson's consonant survey are shown in 1.

Of course, factors such as the surveyor's interpretation of specific sounds when putting them into phone categories could influence what this graph looks like. Therefore, this project does not try to recreate patterns using 'phonemes' like /m/ or /k/ - since typological surveys often have to make assumptions or extrapolations when characterizing a heard sound as a phoneme. Instead, this project focuses on surface-level sounds, or phones, defined in terms of certain articulatory and perceptual features. Strings of 'placeholder phonemes' are used to make the results more interpretable - this is discussed in section 4.4.

Next, we consider sequential constraints. Of particular interest to this project is syllable structure, since it is an easily observable, descriptive feature of a phonology. (Clements, 1990) describes a 'sonority cycle principle' which introduces the idea of sonority peaks in the nucleus of a syllable and the sonority of sounds in a syllable gradually increasing then slightly decreasing. We include sonority as a feature of phones in our study so that syllable structure may be studied in these terms.

Modeling Phonology through Phonetics This section describes some frameworks and experiments that approach the basic evolution of a phonology from a phonetic or acoustic perspective. Only these studies are mentioned here because the approach of this project is directed at sounds at the surface level, not dependent on any particular theory of phonology. Although some of the following studies do draw from specific traditions in phonology such as Optimality Theory (Prince & Smolensky, 2002), the ideas we adopt from them are largely theory-independent.

One approach is developing a theoretical model of phonology in terms of phonetics, and using this to make predictions that can be verified through typological surveys such as the ones referenced in the above subsection. (Liljencrants & Lindblom, 1972) do this with vowels, proposing dispersion as a driving factor in vowel inventories. This refers to the idea that the vowels that are constrastive in a given language would be as far from each other in perceptual space as possible. Liljencrants and Lindblom test their hypothesis with languages having various sizes of vowel inventories, from 3 to 7, and find a general correspondence. This sort of study in general seems to be carried out more with vowels than consonants, since vowels are easier to quantify and place in perceptual space with clear formants, height, backness, etc.

After this, in modeling phonology from a phonetic perspective, a persistent idea is that there are opposing weights of wanting to place your phonemes far apart in perceptual space, and minimizing the articulatory effort of switching between phonemes while doing so. This potential tradeoff leads to an interesting modeling problem, similar in some ways to the information bottleneck in developing semantic spaces (for an example, see (Zaslavsky, Kemp, Regier, & Tishby, 2018)). (Flemming, 2001) applies this to an Optimality Theory framework, by suggesting constraints including MinimizeEffort and MaximizeContrast, which are weighted rather than ranked to create more of a gradient between possible optimal tradeoffs.

There have been attempts to actually run simulations that treat phonemes as entities in perceptual and acoustic space. Some early papers had computational agents pass single vowels to each other, where the goal of the agents was to imitate each other as well as they could and keep an evolutionary advantage (Glotin 1995, as cited by De Boer 2000). (De Boer, 2000) expands significantly upon these initial efforts, equipping a group of agents with models of human perception and articulation for vowels. De Boer's experiments find natural-language-like patterns arising in the phonology evolved collectively by these agents.

De Boer's methodology is meticulous in that it uses well-accepted models of perception and articulation. Interestingly, though, the agents in De Boer's simulation operate with the main goal of imitating each other as well as possible. Intuitively, as language evolves, we expect a balance between wanting to converge to a set of representations (i.e. imitate each other) and wanting to communicate about the world. Importantly, the modeling simulations of phonology thus far do not explicitly tie model utterances to a task or an external world. It is likely that in the case of human language, it is influenced by the reasons for which we communicate - for instance, collaborating on tasks, or communicating ideas. For this reason, studies of agents communicating about events or environments external to themselves is of interest in modeling natural language.

Emergent Communication There is a recent emerging paradigm of experimentation in computer science in which multiple neural network agents may 'communicate' with each other to achieve interrelated goals. This often takes place in a reinforcement learning framework, which means that the agents receive different 'rewards' for different outcomes of the experiment, and are trained to maximize their reward – so, ideally, they learn the optimal policy for maximizing the reward.

In the case of emergent communication, the agents might have the same goal or may have to compete with each other to achieve different goals. Still, the rewards that the agents receive are interdependent. Additionally, sometimes the abilities of the agents are also different (for instance, one agent may have the ability to observe the environment while only the other agent has the ability to actually take actions in the environment). Due to this, the agents are induced to converge upon a meaningful 'communication' system over the course of the training. This is called emergent communication (henceforth, EC).

Emergent communication in multi-agent reinforcement learning settings has recently been studied for giving rise to apparent natural-language-like patterns, as observed in the context of referential games (Lazaridou, Peysakhovich, & Baroni, 2016).Outside of the application of EC to robotics and computer science, we argue it is interesting to linguists for two reasons.

First, there is the question of whether applying natural-language-processing techniques or insights from from human language production and perception experiments can improve this communication between agents. Improvement consists of making the inter-agent communication more efficient, quick to learn, human-interpretable, or robust. A couple of recent effort in this direction have been introducing multidimensional semantic spaces (such as word embeddings) to discrete emergent communication (Tucker et al., 2021) or constraining the communication through making the communication channel noisy, as it often is in the real world leading to inference by rational speaker and listeners.

The second potential contribution of EC is that probing the nature of the communication between neural network agents to compare it to natural-language often shows the differences between the two, the pitfalls of the model, and the ways to induce EC to be more natural-language-like (Kottur, Moura, Lee, & Batra, 2017). Being able to run a semi-controlled experiment like this could be an interesting modeling tool in addition to current theoretical frameworks and rule-based models of linguistics.

This framework is interesting for studying the evolution of phonology since the agents have the incentive to converge on an efficient communication pattern. If the task is simply the reconstruction of an input (i.e. one agent sees the input and emits communication, the other agent decodes from the communication), the agents are incentivized to converge on a vocabulary. Since the phonology is best studied at the level of creating words, the contents of such a vocabulary are ideal to model emerging phonology.

4 Software Architecture

The model architecture essentially includes two collaborating agents, coded as part of a single neural network, operating within an environment that selects inputs and determines rewards based on the the agents' outputs.

4.1 Environment and Task

We define and code a simple 'environment' for the agents where the goal is to essentially replicate one-dimensional vectors. Together, the two agents work as an 'auto-encoder': a neural network architecture whose goal is to encode an input into a latent representation and then decode back into the original input, essentially compressing or converting the information received in the input in a way that it can be recovered.

The environment is initialized with a concept pool C, containing a set number of 'concepts', where each concept is a distinct 1D vector of length `input_dim` that consists of 1s and 0s in some order. For instance, if the variable `input_dim` was set to 6, some examples of inputs that would

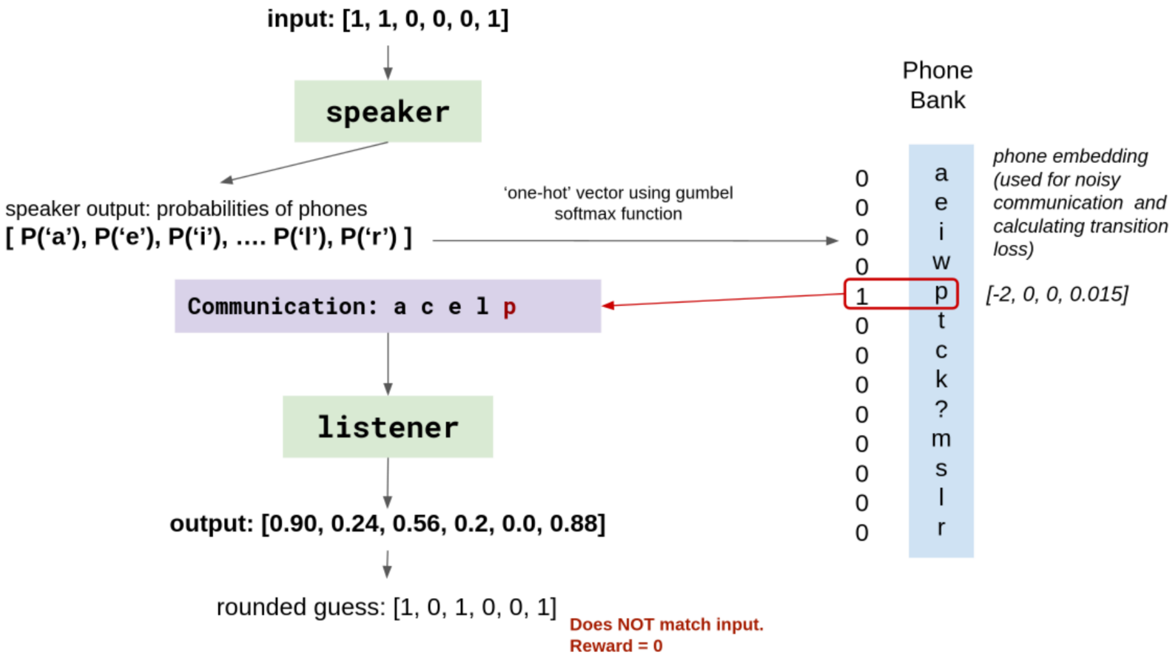


Figure 2: An example of the speaker and listener working during a single episode.

be in C could be [1, 0, 0, 1, 0, 0], or [0, 1, 1, 0, 0, 1], and so on. The size of C is set at the beginning, and then the vectors are randomly chosen until C is full.

The agents themselves are two neural networks referred to here as a `speaker` and `listener`. The training occurs in 'episodes'. In each episode, a single input concept c_i is drawn, which is visible to the `speaker` agent. The speaker sends communication of a fixed length to the `listener`, using which the `listener` guesses a concept $\hat{c}_i$. Both agents get a positive reward if $\hat{c}_i = c_i$ and a reward of 0 if $\hat{c}_i \neq c_i$. The agents are therefore functioning as the encoder and decoder respectively, where the speaker must convert the input into a 'latent representation' in the form of a communication to the listener, and the listener must learn to decode the initial input vector from this latent representation. `speaker` to `listener`. The entire process is illustrated with an example in Figure 2.

This auto-encoder environment was chosen over a reference game environment, in which the `speaker` would see a target image and the `listener` would be required to choose between a target and distractor image. The reference game was the initial choice for this experiment, but it could lead to confounding factors like the emergence of compositionality or communication about features of the input. In a reference game, the target and distractor concepts typically are drawn from different 'categories,' where concepts from a single category have similar features. For instance, a reference game involving images could have categories such as animals, vehicles, etc. and the target and distractor images would always be chosen from different categories. This can lead to models learning an interesting semantic structure, but it is not the focus of this experiment. In human language, breaking down meaning to the level of individual phonemes is rare if even possible, so we are intentionally avoiding it here. The auto-encoder task incentivizes the models to develop a vocabulary that assigns distinct, potentially unrelated communication sequences to each concept in C. We can study the phonological patterns in this vocabulary without thinking about the meaning of the words.

4.2 Models and Software

The model architecture is adapted from the reference implementation for `ic3net` (Singh, Jain, & Sukhbaatar, 2018).[1], modified for phoneme-sequence communication. The `speaker` and the `listener` are two Multi-Layer Perceptron (MLP) models, which essentially transform the input through a series of matrix multiplications, and continuously update the weights for these transformations during training so as to improve performance. Each MLP consists of a linear layer that transforms the input into a latent representation, two hidden linear layers that use the ReLU activation function, and an output layer.

[1] https://doi.org/10.48550/arxiv.1812.09755 Code available under the MIT License.

The architectures of the speaker and listener are described below, and shown in Figure 3 and Figure 4.

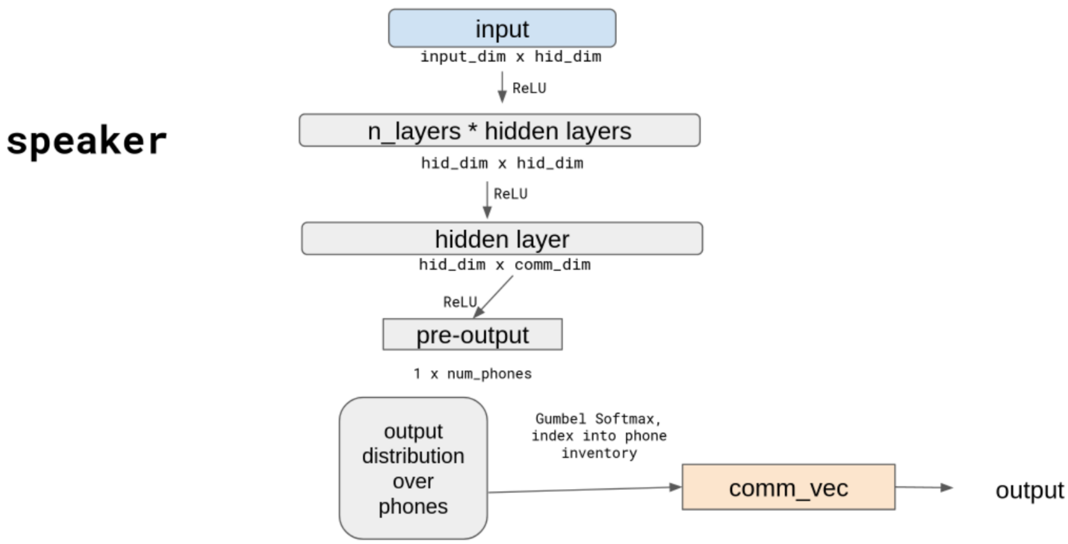
For the speaker, the input is the input concept vector c_i chosen by the environment, and the output at every time-step is a probability distribution over the phones in the phone inventory. The output of the `speaker` is converted into a single chosen phone by applying the `gumbel Softmax` function to it. This function outputs a one-hot vector that has a value of 1 at the index of the chosen phone (and 0 everywhere else), and can be used to select the correct phone embedding from the inventory.

One version of the model implemented communication through a noisy channel. Gaussian noise with a pre-set standard deviation is added to the embedding for the phone selected by `speaker`. The result is then compared to all the phone embeddings to find the closest one in terms of minimum squared distance. This closest phone is added to the final communication passed to the listener. Noise can be turned off or varied.

The listener, similarly to the speaker, has multiple linear layers that transform the current state of the communication vector into a latent representation. Based on this representation, the listener outputs a vector of length `input_dim`. The listener's output layer goes through a Sigmoid activation function, which classifies each of these input_dim positions into a probability between 0 and 1. When calculating the reward, the output of the model is rounded to give a vector of 0s and 1s, which can be compared to the original input.

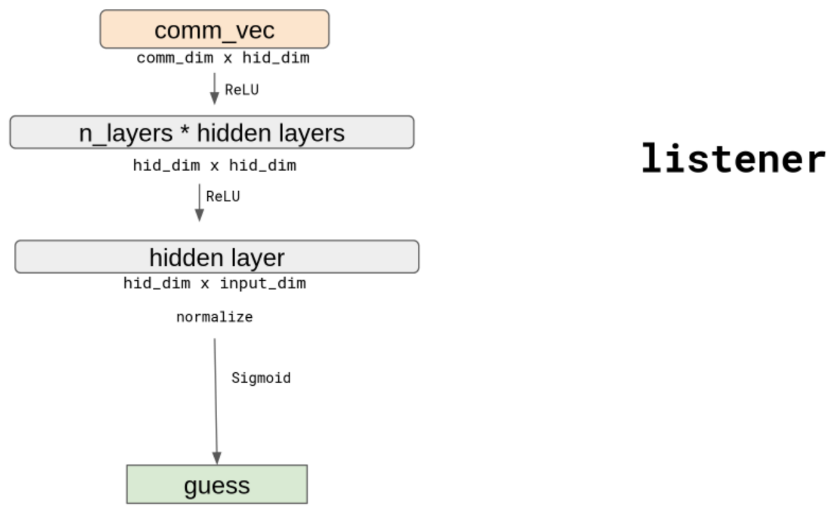
4.3 Reward Function

Apart from the default the reward function that assigns a positive reward for correct guess and zero for incorrect guess, a few different reward functions were tried. One experiment built up



`Comm Vec at step number 4 is tensor([1., 2., 4., 5., 0.]) and the word is eipta.`

Figure 3: Technical diagram of the speaker architecture, with an example output generated by the model.




```
Listener activations are tensor([0.3135, 0.3668, 0.5736]) and its guess is tensor([0., 0., 1.]).
```

Figure 4: Technical diagram of the listener architecture, with an example output generated by the model.

the word step by step, so that the speaker emitted one phone per time step until the full word was formed. The listener could decide at each time step whether to guess or not, except the last time-step which forced a guess. Both agents would receive a large positive reward if a correct guess was made, zero or a negative reward for a wrong guess (there was slightly better performance with zero reward for incorrect guesses), and a small positive reward if the listener took no action at one of the intermediate time-steps.

The motivation for this sort of reward function is human behavioural studies showing people regularly inferring words before hearing them completely. It seems that in natural language communication, as we hear each sound, we are narrowing down the state space until we have a confident guess of the word that the speaker intended. Neural network agents doing the same as they converge to a vocabulary would be an ideal result.

4.4 Representing Phones as Vectors

One of the key challenges of this project is the representation of phones in a way that indicates their acoustic and articulatory characteristics. Most theoretical models of phonology treat phonemes as collections of categorical features. Although this representation lends itself well to a vector format, an ideal representation encodes phones in perceptual (or acoustic) and articulatory space in a more general way.

Eventually, each phone was encoded as a one-dimensional array of key articulatory features. This representation was chosen after reviewing literature in acoustics, as well as through studying spectrograms of the particular sounds of interest.

The phone representations are listed in Table 1. It is important to note that the characters representing phones are simply representations of a set of articulatory features. There is a

	Sonority	Tube Length	Tongue Height	Delayed Release
a	2	1	0	-1
e	2	-1	1	-1
i	2	0	2	-1
w	2	3	2	-1
p	-2	0	0	0.015
t	-2	1	0	0.025
c	-2	2	0	0.030
k	-2	3	0	0.035
?	-2	4	0	0.060
m	0	0	0	-1
s	-1	1	1	-1
l	1	1.3	0	-1
r	1	1.3	1	-1

Table 1: Phone Representations with human-interpretable placeholders and four articulatory-acoustic features.

correlation, but for instance, an experimental result that shows a phone representing a vowel being often selected after one of the phones representing a stop consonant is not necessarily an indication that a particular sequence like 'pa' or 'ti' is favored, but rather that a heavily sonorant, voiced phone is favored after an obstruent with certain features.

5 Discussion and Future Work

5.1 Proposed Experiments

In order to make experimentation easier, the model also outputs a human-interpretable 'word' at every episode of the training, consisting of the phones that it chose for the communication. Some examples are in 5. If the model is able to converge on a vocabulary, these word lists can be used to plot the frequencies of particular segments or sequences.Two key experiments are proposed as the next step once the model learns to converge on a vocabulary.

```
['ekilk', 'ipsmi', 'ekmie', '?kcwp', 'kpr?t', 'pweci',
 'mwrik', 'stwam', 'karrk', 'milka', '?e?ae', 'ewiee',
 's?ilc', 'kakci', 'rcskk', 'awepk', 'etati', 'tcsra',
 'ltlsk', 'elsts', 'pkpee', 'ersw?', '?tail', 'risil',
 'pck?w', 'iwmkp', 'slpll', 'tttk?', 'wlmrt', 'spsta',
 'wicas', 'lrmmr', 'rwwc?', 'eepsa', 'eeais', 'pttrl',
 'sis?w', 'eakii', 'wrcks', 'accpa', 'rcclp', '?ipcm',
 'lelpt', 'ew?cr', 'tkwip', 'msks?', 'mawck', 'tca?e',
 'ckssr', 'm?sem', 'msei?', 'asiwe', 'trmem', 'spc?s',
 'mi?ia', 'cweae', '?str?', 'skacm', 'mitrw', '?klsa',
```

Figure 5: Extract of a word list from the model output.

Experiment 1: Segment Inventory This experiment seeks to establish what phones turn into contrastive segments in the vocabulary. Currently, the distribution over phones that are used in speaker output communications is simply a uniform distribution. This would be expected to change as noise is added (currently, it does not, but that is likely due to an issue in the software architecture). The proposed experiment consists of steadily increasing the noise being applied to the phone embeddings and analyzing resulting phone distributions in the output. The hypothesized result is that a couple of vowels and stop consonants would be the most likely to be represented, with the individual vowels and stop consonants being far from each other perceptually.

Experiment 2: Sequential Constraints Section 5.2 will talk about `transition_loss`, which is a penalty applied to transitions between phones which are difficult for articulation (for instance, successive phones being far from each other in place of articulation). `transition loss` is currently not applied in the default model, for simplification. We can see in Figure 5 that there seem to be no constraint on how phones are put together. Upon adding this loss, the hypothesized result is that phones closer together in location and manner of articulation will be frequently emitted together. In this experiment, constraining the number of vowels in the base phone inventory can be used to probe the effects of a sonority sequencing/cycling phenomenon.

5.2 Results

Progress and Contributions This project builds the basic architecture required to simulate phonology through emergent communication. In particular, the model encoder successfully outputs a selection of phonemes, either all at once or over consecutive time-steps, and the model decoder converts this into a guess about the contents of the input. The environment correctly assigns a reward based on the decoding.

The code developed in this project includes additional arguments that control sections of the architecture, relevant in the proposed experiments. Additionally, individual variables in the code can be updated to study their effect on the system.

- `full_words` : This controls whether the speaker must communicate in full words, or outputs a single phoneme at every time step until the full word is created. `full_words` is currently on by default, which means the listener must wait for the whole word before guessing the input. If this variable is turned off, the listener can choose whether or not to guess at every time step depending on its confidence level. In that case, the listener's output is a series of `input_dim` + 1 probabilities, where the first probability corresponds to confidence level and the remainder is the listener's guess.

- `transition_loss` : This is a loss penalizing articulatory constraints. Specifically, we propose using the mean squared error over the phone embeddings since sounds that are articulated far apart have values far from each other, and also correspond to higher effort in order to articulate them one after the other. Used together with noise, the `transition_loss` can be altered to study the tradeoff between perceptual distance and ease of articulation.

Shortcomings and Modifications The key shortcoming is that the current version of the model fails to converge to a shared vocabulary between the agents. Learning a shared vocabulary is a crucial first step to carrying out the experiments proposed above, and probably the trickiest one.

A possible reason for the model failing to converge is some non-differentiable transformation in the process of getting from input to reward. This was probed by simplifying the model, removing transition loss entirely as well as removing noise. `gumbel softmax` is used in the final activation step of the speaker model since introducing an `argmax` function to retrieve the phone index brought in a non-differentiable transformation.

The failure is likely not due to the reward function, since multiple different reward functions were tried. The model was trained with a variety of hyperparameters (num_epochs between 10 to 50, epoch_size between 20 to 100, batch_size 1).

5.3 Future Work

The clear next step for this work is to make architectural changes in the simulation code and see if the models can converge on a vocabulary. If there is no convergence, it is possible that the task itself will have to be changed - from an auto-encoding task to something else.

There are also ways to expand upon the current proposed experiments, using this or a similar paradigm. One example is varying the length of the communication between agents (throughout the examples in this paper, that length is 5) and observing the effects on the two proposed experiments.

The most clear thing that can be changed is the phone embeddings. Currently, the phones are a single vector that represents acoustic-articulatory features. In a future experiment, it may be interesting to incorporate a model of human perception such as the Maeda model for vowel simulation (Maeda, 1982) to convert these articulatory features directly into acoustic or perceptual ones. If a similar model is not available for consonants, it may be possible to do experiments with vowel inventories, similar to (Liljencrants & Lindblom, 1972) but with a computer model rather than a theoretical framework.

References

Clements, G. N. (1990). The role of the sonority cycle in core syllabification. In J. Kingston & M. E. Beckman (Eds.), *Papers in laboratory phonology* (Vol. 1, p. 283–333). Cambridge University Press. doi: 10.1017/CBO9780511627736.017

Crothers, J. H., Lorentz, J. P., Sherman, D. A., & Vihman, M. M. (1979). *Handbook of phonological data from a sample of the world's languages: A report of the stanford phonology archive.*

De Boer, B. (2000). Self-organization in vowel systems. *Journal of Phonetics*, *27*, 1-25.

Flemming, E. (2001). Scalar and categorical phenomena in a unified model of phonetics and phonology. *Phonology*, *18*.

Hyman, L. (2008, 04). Universals in phonology. *Linguistic Review - LINGUIST REV*, *25*, 83-137. doi: 10.1515/TLIR.2008.003

Kottur, S., Moura, J. M. F., Lee, S., & Batra, D. (2017). *Natural language does not emerge 'naturally' in multi-agent dialog.* arXiv. Retrieved from `https://arxiv.org/abs/1706.08502` doi: 10.48550/ARXIV.1706.08502

Lazaridou, A., Peysakhovich, A., & Baroni, M. (2016). *Multi-agent cooperation and the emergence of (natural) language.*

Liljencrants, J., & Lindblom, B. (1972). Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language*, *48*(4), 839–862. Retrieved 2022-05-11, from http://www.jstor.org/stable/411991

Maddieson, I. (1984). *Pattern of sounds*. Cambridge, UK: Cambridge University Press.

Maeda, S. (1982). A digital simulation method of the vocal-tract system. *Speech Communication*, *1*(3), 199-229. Retrieved from https://www.sciencedirect.com/science/article/pii/0167639382900176 doi: https://doi.org/10.1016/0167-6393(82)90017-6

Prince, A. S., & Smolensky, P. (2002). Optimality theory: Constraint interaction in generative grammar.
doi: 10.7282/t34m92mv

Singh, A., Jain, T., & Sukhbaatar, S. (2018). *Learning when to communicate at scale in multiagent cooperative and competitive tasks.* arXiv. Retrieved from https://arxiv.org/abs/1812.09755 doi: 10.48550/ARXIV.1812.09755

Tucker, M., Li, H., Agrawal, S., Hughes, D., Sycara, K. P., Lewis, M., & Shah, J. (2021). Emergent discrete communication in semantic spaces. *CoRR*, *abs/2108.01828*. Retrieved from https://arxiv.org/abs/2108.01828

Zaslavsky, N., Kemp, C., Regier, T., & Tishby, N. (2018). Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, *115*(31), 7937-7942. Retrieved from https://www.pnas.org/doi/abs/10.1073/pnas.1800521115 doi: 10.1073/pnas.1800521115